

# ***Creació d'un bot d'assistència a la resolució d'incidències.***

Autor: Pol Chica Capell

Director: César Hernández Baigorri, Technology 2 Client

Ponent: Lluís Padró, Departament de Ciències de la Computació

Especialitat: Enginyeria de computació

Facultat de Informàtica de Barcelona

Universitat Politècnica de Catalunya

Technology 2 Client

23 de Octubre de 2018



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA  
BARCELONATECH



## Resum

Els agents conversacionals o chatbots son una branca del Processat de Llenguatge Natural (NLP) que actualment esta despertant gran interès, sobretot en els àmbits de atenció al client, FAQs i sistemes de assistència personal, com serien Siri o Cortana.

Aquest projecte, proposa la implementació de un sistema que utilitzant un agent conversacional, sigui capaç de solucionar incidències i de proporcionar informació personalitzada al client. El sistema disposa de una estructura client servidor que es comunica mitjançant una api rest. En la part del servidor, s'ha implementat un chatbot que mitjançant una sèrie de normes es capaç de identificar casos determinats, i amb un model entrenat, es capaç de adaptar-se a la resta de situacions.

L'objectiu de aquest projecte es realitzar una aplicació que implementant un agent conversacional, sigui capaç de aportar solucions alhora que pot proporcionar informació. Addicionalment, amb el projecte es pretén realitzar una solució personalitzada a les alternatives ja existents en el mon del chatbots.

## Resumen

Los agentes conversacionales o chatbots son una rama del Procesado de Lenguaje Natural (NLP), que en estos momentos está suscitando mucho interés, sobre todo en los campos de la atención al cliente, FAQs y sistemas de asistencia personal, como serian Siri o Cortana

Este proyecto, propone la implementación de un sistema que, mediante el uso de un agente conversacional, sea capaz de solucionar incidencias y proporcionar información personalizada al cliente. El sistema consta de una estructura cliente servidor que se comunica mediante una api rest, este sistema permite una mayor accesibilidad a la herramienta. En la parte del servidor, se ha implementado un chatbot que, usando una serie de normas, es capaz de identificar una serie de casos determinados y aportar una solución, y mediante el uso de un modelo entrenado, sea capaz de adaptarse al resto de situaciones.

El objetivo de este proyecto, es realizar una aplicación que, con la implementación de un agente conversacional, sea capaz de aportar soluciones a la vez que puede proporcionar información. Adicionalmente, con el proyecto se pretende realizar una solución personalizada a las alternativas ya existentes en el mundo de los chatbots.

# Abstract

Conversational agents or chatbots are a branch of Natural Language Processing (NLP), which is currently raising a lot of interest, especially in fields like customer service, FAQs and personal assistance systems, such as Siri or Cortana.

This project proposes the implementation of a system that, through the use of a conversational agent, is able to solve issues and providing personalized information to the client. The system consists of a server-client structure that communicates through an api rest, this system allows greater accessibility to the tool.

On the server side, a chatbot has been implemented, this is able to identify an amount of specific cases and provide a solution by using a group of rules, and by using a trained model, is able to adapt to the rest of situations.

The goal of this project is to make an application that, with the implementation of a conversational agent, is able to provide solutions while giving information. Additionally, the project focuses on working in a customized solution to existing alternatives in the world of chatbots.

## Agraïments

Aquest projecte, sense dubte, ha implicat a gran quantitat de persones, desde les persones que han estat des del primer moment, en el que el projecte encara era només una idea poc clara i sense un rumb clar, fins a les que s'han incorporat a mig camí o de forma esporàdica. Totes elles, han aportat la seva part per a que el projecte hagi pogut concloure amb èxit.

Per part de la universitat, agrair al meu ponent, Lluís Padró, per la seva col·laboració i la seva ajuda encaminar el projecte en la millor direcció possible.

Per part de la empresa, agrair al director, Cèsar Hernández, per tot el suport, ajuda aportada en el disseny i entregues, i sobretot, per haver estat a sobre evitant possibles desastres. A Xavier Rodríguez, Lionel Magan i Christian Romero, per la seva incalculable ajuda en el desenvolupament del projecte, y a Enric García, per donar-me la possibilitat de realitzar aquest projecte a la empresa. També agrair el suport a la resta de companys que han estat disposats a ajudar dia a dia.

També m'agradaria donar les gràcies a tots els que m'han donat suport al llarg del projecte fora de l'aspecte acadèmic. Agrair als meus amics per tots els bons moments que m'han permès desconnectar en moments d'estres. A la meua família, per haver-me suportat i recolzat al llarg del projecte. Però sobretot, hi ha una persona a qui voldria donar les gràcies, perquè sense ella el projecte no hauria arribat fins a on ha arribat, per tot el suport que m'ha donat i perquè ha fet que tot fos molt més fàcil.

Resum	1
Resumen	2
Abstract	3
Agraïments	4
1. Introducció	9
1.1. Formulació del problema	9
1.2. Objectius del projecte	11
1.2.1. Anàlisi de incidències	11
1.2.2. Entorn e infraestructura del projecte	12
1.2.3. Chatbot	12
1.2.4. Client web	12
1.3. Contextualització	13
1.4. Actors implicats	13
Desenvolupador i dissenyador	13
Director del projecte	13
Ponent del projecte	13
DevOps	14
Usuaris	14
Professor de GEP	14
Empleats de la empresa	14
Beneficiaris	14
2. Estat de l'art	15
2.1. Conceptes sobre llenguatge natural i xarxes neuronals	15
Coneixements sobre NLP	15
Coneixements sobre Neural Networks	15
Coneixements sobre el model encode/decode	16
2.2. Chatbots	16
2.3. Chatbots en la actualitat	17
2.3.1. Chatbots basats en regles o comandes	18
2.3.2. Chatbots basats en IA	18
2.4. Estat del mercat	18
2.4.1. Chatbots per a la interacció amb clients	18
2.4.2. Chatbots per a optimitzar tasques i processos	18
2.4.3. Chatbots en xarxes socials	19
2.4.4. Chatbots per a augmentar les ventes	19
2.5. Conclusió	19

3.	Definició de l'abast	20
3.1.	Abast	20
3.2.	Possibles obstacles	20
	Errors en el codi	20
	Calendari	21
	Dificultat en la automatització de incidències	21
3.3.	Metodologia	21
3.4.	Mètode de avaluació	21
3.5.	Eines per al desenvolupament	21
	Python (3.6)	22
	MySQL	22
	HTML,CSS	22
	Tensorflow	22
	Keras	22
	Spacy	22
	Insomnia	22
4.	Fases del desenvolupament	23
4.1.	Resum de les fases	23
4.2.	Detalls de les fases del desenvolupament	24
4.2.1.	Preparació	24
	Planificació	24
	Estudi de aplicabilitat	24
	Estudi de la tecnologia	24
	Preparació de l'entorn	24
4.2.2.	Execució	24
	Automatització de les aplicacions	24
	Implementació del chatbot	25
	Implementació del client web	25
4.2.3.	Avaluació	25
	Comprovació dels resultats obtinguts	25
4.2.4.	Go-Live	25
	Test amb els usuaris	25
4.2.5.	Memòria	25
	Redacció de la memòria i resultats	25
	Formació al equip	25
4.3.	Valoració d'alternatives	26

4.4.	Diagrama de Gantt	27
4.5.	Integració dels coneixements	28
4.6.	Identificació de lleis i regulacions	29
5.	Gestió econòmica i sostenibilitat	30
5.1.	Gestió econòmica	30
5.1.1.	Consideracions i comentaris	30
5.1.2.	Pressupost de recursos humans	30
5.1.3.	Pressupost de hardware	31
5.1.4.	Pressupost de software i llicències	31
5.1.5.	Despeses indirectes	31
5.1.6.	Pressupost total	32
5.1.7.	Control de desviacions	32
5.2.	Àrea social	33
5.3.	Àrea ambiental	33
5.4.	Puntuació sostenibilitat	33
5.5.	Conclusió sobre la sostenibilitat	34
6.	Disseny	35
6.1.	Detalls de la infraestructura dissenyada	35
6.1.1.	Descripció de la infraestructura	35
	Node servidor	35
	Node de base de dades	36
	Node client	36
6.1.2.	Descripció de les tecnologies potencials	36
6.2.	Disseny del chatbot	37
7.	Implementació	38
7.1.	Entorn	38
7.2.	Instal·lacions i configuracions inicials	38
7.2.1.	Instal·lació del sistema operatiu	38
7.2.2.	Instal·lació de python	38
7.2.3.	Instal·lació de llibreries	39
7.3.	Automatització de incidències	39
7.3.1.	Obtenció de informació	40
7.3.2.	Resolució de incidències	41
7.4.	Implementació del chatbot	41
7.4.1.	Primera capa: Arbre de decisions	42
	Preprocessament de la entrada	42



Presa de decisions	43
7.4.2. Segona capa: seq2seq	43
Entrenament	43
Predicció	45
7.5. Implementació de la interfase web	45
7.6. Comunicació web-chatbot	47
8. Conclusions	48
9. Treball futur	49
Referències	50

# 1. Introducció

## 1.1. Formulació del problema

Els avenços tecnològics dels últims anys han permès que tasques que abans es creien adients únicament per un recurs humà siguin avui dia automatitzables en gran mesura o es pugui augmentar les capacitats del humà. Això permet, amb menys recursos, donar un alt grau de servei i deixar que el talent s'enfoqui a problemes de coneixement i no repetitius.

Un dels camps que, malgrat existir a nivell conceptual desde els anys 50 [\[2\]](#), ha permès aquest gran desenvolupament ha estat les xarxes neuronals. Aquestes sumades al avenç en tecnologia de xarxes, emmagatzematge i processat ha permès que avui dia pugui tractar-se un gran volum de dades en un temps raonable. De tal manera que les dades encara son vàlides quan arribem al resultat. Descodificar el genoma humà va ser una tasca de 10 anys de processat quan desde que es va iniciar al 2003 [\[20\]](#) [\[21\]](#) , avui dia pot fer-se en una setmana.[\[22\]](#)

Una de les aplicacions més prominents avui dia es l'ús de chatbots. Aquets permeten extreure entitats arrel de llenguatge natural, el que comunament es denomina com a NLP. Aquest projecte pretén fer us d'aquesta tecnologia per millorar la qualitat de servei oferida al client i deixar que els consultors es centrin en tasques de coneixement que aportin valor a l'empresa.

Els centres d'atenció al client són un dels recursos més comuns en les grans empreses de gestió de serveis. A més, és un dels llocs on es destinen més recursos humans dins d'una empresa. Les tasques que desenvolupen aquests centres d'atenció al client poden ser molt diferents, varien en funció del tipus de client a qui es vulgui atendre i del tipus de servei ofert.

En el nostre cas, ens trobem en una situació d'una empresa que gestiona serveis per a clients (manteniment IT). Entre el que la empresa ofereix al client, s'inclou una gestió

de incidències, que també implica solucionar-les. Es per això que es requereix d'un equip completament dedicat a aquesta tasca. Aquest equip, ha de proporcionar una atenció continuada al client, i freqüentment, realitzant tasques de poca afectació, o que podrien ser fàcilment automatitzables.

Per això, es detecta un problema en la destinació dels recursos de l'equip, ja que el temps de la gent de l'equip de resolució de incidències podria dedicar-se a realitzar tasques o projectes que puguin aportar més valor a la empresa.

## 1.2. Objectius del projecte

L'objectiu final d'aquest projecte és crear una eina que permeti ajudar als tècnics del centre d'atenció al client a reduir el nivell d'atenció sobre algunes de les sol·licituds que reben. Això últim permetria que es centressin en tasques que aportin valor i no siguin automatitzables, donant així una resposta més acurada a aquelles sol·licituds que ho requereixen o utilitzant la capacitat sobrant de l'equip per desenvolupar millores en comptes de centrar-se tan intensivament en el manteniment i l'atenció al usuari.

El *chatbot* com a eina ja es fa servir actualment en molts centres d'atenció al client, de totes maneres una gran quantitat d'aquests limiten el seu abast al punt d'aportar guies o solucions per indicar al usuari que ha de fer en segons quines situacions. Aquest projecte pretén diferenciar-se anant un pas més enllà, mitjançant scripting pretenem no només servir com a repositori d'informació per l'usuari final sinó ser també capaç d'executar processos que resolguin situacions amb una alta ocurrència i fàcilment automatitzables.

Un cop establert l'objectiu d'aquest projecte s'han marcat les següents fites a assolir durant la seva realització per tal d'arribar a una solució final que aportí valor a l'empresa:

### 1.2.1. Anàlisi de incidències

Amb l'estudi de les incidències rebudes s'aconsegueix:

- Analitzar les més freqüents,
- Aïllar les que puguin ser solucionades de forma automàtica,
- detallar-ne els mètodes de resolució i trobar la forma de diferenciar-les d'altres incidències
- implementar la forma de solucionar la incidència de forma automàtica.

Aquest objectiu és el que permetria alleugerir en gran part la feina del equip de sistemes.

### 1.2.2. Entorn e infraestructura del projecte

Per poder fer que l'eina sigui el més accessible possible per a l'usuari es dissenya de tal forma que sigui adaptable a la majoria de sistemes i pugui funcionar amb la màxima autonomia possible. Per això s'implementa una eina que sigui independent del sistema operatiu. En quant a la estructura del projecte, disposa de 3 nodes principals, el primer consistiria en el node de base de dades, el segon el node on la aplicació està executant-se, i el últim seria la part del client que accedeix al node del servidor. Aquesta estructura permet accessibilitat des de qualsevol lloc a la eina, facilita el manteniment de les dades i que la estructura sigui escalable.

### 1.2.3. Chatbot

Per a l'eina s'implementa un chatbot, que disposa de les següents característiques:

- Habilitat per a respondre peticions definides
- Capacitat per a interactuar amb l'usuari
- Facilitat per redissenyar les peticions que es capaç de tractar el chatbot

### 1.2.4. Client web

Per tal de poder interactuar amb el chatbot s'implementa un client web. Aquest client permet comunicar-se amb el bot establint canals únics de comunicació. Sent accessible des de qualsevol navegador compatible i amb un usuari i contrasenya vàlid. En un entregable final, s'hauria d'incorporar a l'eina de missatgeria del client per a que tots els usuaris el tinguin a l'abast.

### 1.3. Contextualització

Aquest problema va més enllà, sent un pas endavant en la creació d'un servei d'atenció al client automatitzat que facilita el contacte entre les empreses i els clients. L'automatització d'aquest recurs s'ha dut a terme de diverses formes, amb softwares de trucades automàtiques, assistents virtuals, etc.

Tot i aquestes eines existents, en voler crear un sistema de resolució d'incidències automàtic es requeria una solució personalitzada a partir d'un estudi intern de les estructures IT, per tant, la millor opció era realitzar l'eina de forma interna.

### 1.4. Actors implicats

Els actors implicats en el projecte, és a dir, aquelles persones que poden estar interessades pel projecte o se'n poden beneficiar d'ell, són els següents:

#### Desenvolupador i dissenyador

Les tasques de desenvolupament i disseny del projecte seran realitzades per mi mateix. Aquestes inclouen a més de la programació dels diversos components del projecte també l'elaboració de la memòria. Per això compta amb l'ajuda dels següents actors.

#### Director del projecte

El director de l'empresa és en César Hernández Baigorri, gestor de projectes de l'empresa Technology 2 Client. El seu paper és supervisar que es compleixi el calendari estipulat amb el projecte i que s'assoleixin els objectius marcats. A més, pot guiar al desenvolupador a l'hora de realitzar el projecte.

#### Ponent del projecte

El ponent del projecte és Lluís Padró, professor de la UPC i expert en l'àrea de llenguatge natural. El seu paper és el de supervisar el projecte i donar consell tècnic en la seva àrea d'expertesa, a més de fer que el treball s'ajusti a la normativa acadèmica de la facultat.

### DevOps

El personal de suport de la empresa Technology 2 Client. Participen proveint de consells i ajut per al desenvolupament del projecte.

### Usuaris

Els usuaris de l'eina seran, per un costat, els clients que es comunicaran amb l'eina directament per informar de les incidències i, per altre costat, els tècnics als quals ha d'alliberar de càrrega de treball, que hauran d'estar informats sobre les incidències resoltes de forma automàtica.

### Professor de GEP

El tutor per part del mòdul de GEP, en el meu cas, Fco. Xavier Llinars. Durant el mòdul previ al TFG, GEP, es comença a preparar la memòria del projecte. El seu rol és aconsellar sobre bones pràctiques tant a l'hora de redactar documents com al presentar projectes.

### Empleats de la empresa

La resta d'empleats de l'empresa que poden donar suport al projecte. A l'empresa hi ha diverses persones amb coneixements que es poden aplicar en major o menor mesura. La seva ajuda és indispensable pel desenvolupament del projecte.

### Beneficiaris

Els beneficiats per aquest projecte seria l'empresa, ja que la suposada posada a producció del projecte implicaria un estalvi de temps dedicat a resoldre incidències, també el client es veuria afectat per la posada en pràctica del projecte, ja que aquest implica un augment en la velocitat de resolució d'algunes incidències.

## 2. Estat de l'art

### 2.1. Conceptes sobre llenguatge natural i xarxes neuronals

#### Coneixements sobre NLP

El processament de llenguatge natural consisteix en tractar una entrada en llenguatge natural per a tal de que la maquina pugui entendre i interpretar amb el fi de poder mantenir una conversa. Per a això es realitzen diversos tipus de anàlisis, com serien anàlisi morfològic, sintàctic, semàntic i pragmàtic. Amb la conjunció de aquests anàlisis es pot arribar a extreure informació de textos. Les utilitats més conegudes son les següents:

Classificació de documents, mitjançant un classificador de gramàtica, indexat dels mots i detecció de duplicats.

#### Coneixements sobre Neural Networks

Les xarxes neuronals son un tipus de algoritme de Machine learning [\[1\]](#) que imitant el comportament de les neurones del cervell, incorpora un conjunt de neurones artificials que composades de diverses formes, permeten identificar diverses característiques.

El primer model de neurona artificial, va ser proposat per el neuropsicòleg Warren McCulloch i el matemàtic Walter Pitts al 1943 [\[2\]](#). El model, tractava de una aproximació matemàtica sobre una neurona capaç de operar funcions simples com serien una AND, una OR o una NOR.

Tot i partir del model anterior, un dels models més utilitzats es el dissenyat al 1957 per en Frank Rosenblatt, anomenat Perceptró [\[3\]](#). Que a diferencia de l'anterior, que nomes tracta amb valors binaris, aquest es capaç de operar amb números.

L'algoritme que disposa de una funció de activació que rep com a paràmetres la suma amb pesos de les diverses entrades i un *bias*, que dona un grau de llibertat addicional:

$$output = f(\sum_i x_i * w_i + w_o)$$

Els valors òptims per al *bias* i els pesos, es calculen amb tècniques de descens per gradient (Gradien descent).



Amb diversos Perceptrons, es possible generar una xarxa que permet que qualsevol entrada, aquesta s'aconsegueix amb diverses capes de Perceptrons que acaben aconseguint classificar qualsevol entrada.

Aquest sistema, s'anomena MLP (multilayer perceptron) [\[23\]](#) es només un dels tipus que existeixen, depenent del problema es pot escollir una xarxa neuronal capaç de retroalimentar-se.

En el nostre projecte, en específic, seria necessari, un tipus de xarxa neuronal en específic, anomenada LSTM Network (Long Short Term Memory Network) [\[4\]](#) que es tracta de un tipus de xarxa recurrent, es a dir, es capaç de mantenir dades entre entrades, tot per la estructura interna que té.

## Coneixements sobre el model encode/decode

Per a la estructura del projecte, s'ha fet servir la estructura coneguda com a encoder/decoder. Aquesta estructura agafa la entrada de mida variable, i la introdueix en un vector anomenat context, de mida fixe dins de un espai vectorial. Després, el decoder, agafa el vector i el projecta al espai original. Per a entrenar aquest tipus de models, es donen parells entrada resposta, per a minimitzar l'error en la cerca de una resposta. [\[5\]](#) [\[6\]](#)

## 2.2. Chatbots

Actualment, en la nostra societat el desenvolupament de la intel·ligència artificial està cada cop més present a les nostres vides fet que amb el pas de temps, esdevindrà una realitat en el nostre dia a dia.

Centrant-los en l'eina de creació en què basa aquest treball, el *chatbot*, podem esmentar que aquests són utilitzats principalment per dur a terme les funcions d'atenció al client d'una empresa, per la qual cosa són eines capacitades per poder resoldre les necessitats d'una persona. En aquest sentit, podem fer referència al fet que els *chatbots* són una eina informàtica que possibiliten el manteniment d'una

conversa real usuari-màquina en entorns de text o d'àudio, tant a l'hora de demanar-li algun tipus d'informació com de que dugui a terme una acció.

Un dels grans avantatges dels *chatbots* és que, a diferència de les aplicacions no s'han de descarregar, tampoc és necessari actualitzar-los i no ocupen espai en la memòria de l'ordinador. Una altre avantatge que podem trobar és que podem tenir integrats diversos bots en un mateix xat. D'aquesta forma ens evitariem estar saltant d'una app a una altra segons el que necessitem a cada moment.

Els *chatbots* van sorgir per la incorporació per part del centres de trucades de les *IVR* (Respostes de veu interactives) que ajudaven a solucionar aquesta necessitat tant de les empreses com dels clients de donar resposta a les consultes i als problemes dels usuaris.

Més endavant, gràcies als avenços en aquest camp, van sortir el que es coneix avui en dia com a *chatbots*, que a més de substituir les tasques de les *IVR* permetien també realitzar tasques repetitives.

Els chatbots tenen les següents característiques:

- Poden entendre text, tant escrit com parlat e interpretar el seu significat.
- Poden buscar informació i lliurar-la al usuari.
- Son utilitzats per la majoria de aplicacions modernes de telèfons intel·ligents.

## 2.3. Chatbots en la actualitat

Actualment, els *chatbots* s'han desenvolupat fins a arribar al punt de semblar un humà real en la seva interacció amb l'usuari. Inicialment, això no es donava, ja que aquests seguien un arbre lògic que els feia continuar per a poder esbrinar el que requeria l'usuari, això acabava esdevenint en una conversa llarga i rutinària [26]. Tot i haver arribat al punt d'entendre el llenguatge i ser capaços de seguir una conversa real, s'han distingit dues classes de *chatbots*:

### 2.3.1. Chatbots basats en regles o comandes

Es guien per una sèrie de paraules clau que li permeten identificar la intenció de l'usuari. Estan limitats al nombre de respostes que poden respondre, ja que es requereix una programació manual de cadascuna de les accions. Generalment, responen a cerques de plantilles per a la comprensió de les preguntes. [\[24\]](#)

### 2.3.2. Chatbots basats en IA

Poden realitzar un seguiment del context de la conversa, recorden fets d'anteriors converses i són capaços d'aprendre amb ajut de supervisió humana. També responen preguntes ambigües seguint el context de la conversa. A més, també generen respostes. El fet de tenir una forma d'aprenentatge implica que cada cop són més capaços d'oferir resposta i, per tant, més intel·ligents. [\[25\]](#)

## 2.4. Estat del mercat

En la actualitat els chatbots destaquen en 5 sectors:

### 2.4.1. Chatbots per a la interacció amb clients

Aquest tipus de *chatbot* serveix per facilitar a l'usuari la cerca d'informació. Un clar exemple d'aquest tipus serien *Siri* o *Cortana* els dos assistents virtuals d'*Apple* i *Microsoft* respectivament. Són eines que permeten facilitar molt la cerca d'informació en un moment determinat. Aquest tipus d'assistents són utilitzats de forma diària per milers de persones per tal d'estalviar una gran quantitat de temps en cerques.

### 2.4.2. Chatbots per a optimitzar tasques i processos

L'ús de *chatbots* no està només limitat a l'ús de clients de forma interna sinó que també es poden fer servir per a cerques d'informació. Exemples d'aquest tipus de *chatbots* serien *GrowthBot*, una eina que es connecta a bases de dades i sistemes per aportar informació en temps real d'utilitat per al negoci o *MarketSpace* que informa de les últimes novetats d'empreses d'interès i de nous productes, etc.

### 2.4.3. Chatbots en xarxes socials

Existeix un gran nombre de bots en plataformes com *Twitter*, *Instagram*, etc. que permeten de forma automàtica propagar idees i missatges, a part de generar seguidors.

### 2.4.4. Chatbots per a augmentar les ventes

Aquest tipus de bot, es troben en dues formes:

Els primers com a assistent de venda. Aquests bots assisteixen als clients de forma propera i els aconsellen, identificant alhora els gustos i preferències del client.

Els segons sense cap forma d'interacció, simplement recull els productes que cerca l'usuari en una web i actualitza la informació sobre el possible client personalitzant-la per la propera compra suggerint-li productes que puguin cridar-li l'atenció.

## 2.5. Conclusió

Un cop vistos els tipus de *chatbots* que existeixen i els usos que se'ls hi pot donar sembla que la millor solució és crear un *chatbot* híbrid, capaç de seguir una conversa però amb la capacitat d'identificar paraules, a més de, poder seguir un patró determinat per a la incidència pertanyent un cop identificat el problema.

## 3. Definició de l'abast

### 3.1. Abast

Per aconseguir una eina que permeti en major mesura donar resposta i proporcionar una programació de l'eina s'han dut a terme les següents tasques:

Primerament, és necessari realitzar un estudi de quines són les peticions o incidències més freqüents i quines serien capaces de ser tractades de forma automàtica.

Un cop analitzades quines són aquestes incidències, es dissenya el pla d'acció per solucionar cadascuna d'aquestes. Per dur a terme això s'implementarà un *script* específic per cadascuna amb la millor solució al problema. Prèviament, cal seguir una sèrie de passos abans de començar amb la dels procediments.

En paral·lel, en l'anàlisi de les incidències, s'implementarà el *chatbot*, és a dir, la part que es relacionarà amb l'usuari i la lògica de la qual disposa aquest mòdul.

Un cop estiguin ambdues parts implementades es procedirà a vincular les reaccions del *chatbot* amb els corresponents plans d'acció.

### 3.2. Possibles obstacles

Ara analitzarem possibles complicacions que puguin sorgir durant la realització del projecte

#### Errors en el codi

A l'hora de realitzar projectes *software*, un dels errors més freqüents són els de programació. Aquests errors s'han de minimitzar, ja que la seva existència pot produir resultats diferents dels esperats. Per a minimitzar el nombre d'aquests es realitzaran proves a cada versió del codi per verificar que els resultats són els esperats.

## Calendari

Com el temps del projecte és bastant limitat, caldrà fixar un calendari setmanal amb fites estrictes i entregues constants per tal que el projecte no s'endarrereixi.

### Dificultat en la automatització de incidències

És possible que en algunes incidències no sigui possible automatitzar els plans d'acció, fet que podria comportar una pèrdua de temps, recurs força limitat.

## 3.3. Metodologia

El mètode de treball que s'ha seguit correspon al que es denomina metodologia àgil. S'extrau una idea inicial del projecte i es va desenvolupant de forma incremental a mesura es van aportant noves idees. S'estableixen fites per a cada tasca i un cop complertes, s'avalua si aquestes tasques han estat complertes i si no es així, es proposen alternatives.

## 3.4. Mètode de avaluació

El mètode de avaluació es tracta principalment de un estudi funcional, es a dir, es mira que la resposta del *chatbot* sigui la esperada en la situació requerida. Per a això es realitzen tests per a cadascuna de les parts i es comproven els resultats. Finalment es realitza un test global de tot el flow de la aplicació.

## 3.5. Eines per al desenvolupament

Pel desenvolupament del projecte són necessàries diverses eines i llenguatges de programació, les quals es definiran a continuació:

## Python (3.6)

Llenguatge de programació interpretat, orientat a objectes en el qual s'ha realitzat la implementació del chatbot, s'ha escollit aquest llenguatge perquè hi ha gran quantitat de llibreries de *Machine Learning* en aquest llenguatge. [\[9\]](#)

## MySQL

Aplicació de gestió de bases de dades utilitzada per l'enviament de dades entre la web i el servidor de l'aplicació. [\[10\]](#)

## HTML,CSS

Llenguatges utilitzats per la creació del client web que servirà d'interacció client-*chatbot*. [\[11\]](#) [\[12\]](#)

## Tensorflow

Biblioteca de *software* lliure que proporciona eines per treballar amb grans volums de dades. [\[13\]](#)

## Keras

Llibreria que permet treballar de forma intuïtiva amb neural *networks* i presenta compatibilitat amb *Tensorflow*. [\[14\]](#)

## Spacy

Llibreria que permet treballar amb llenguatge natural de forma ràpida. [\[15\]](#)

## Insomnia

Aplicació que permet realitzar crides a *api* de forma intuïtiva. [\[16\]](#)

## 4. Fases del desenvolupament

### 4.1. Resum de les fases

A continuació, es pot veure una taula resumint les fases en les que s'ha dividit el projecte. També es detalla el temps que s'ha invertit en cadascuna de les fases.

FASE	DESCRIPCIÓ	DURADA
1.Preparació	Aquesta fase incorpora les preparacions inicials, com estudis, planificacions, anàlisis i preparacions de entorns	
2.Execució	En aquesta fase es realitzen les implementacions del projecte. Aquesta fase es realitza conjuntament amb la següent, avaluació. En aquesta es realitza la implementació del <i>chatbot</i> i la del servei web	
3.Avaluació	A aquesta fase s'avaluen els resultats de la anterior fase. Si aquesta no es satisfactòria, s'ha de tornar a la fase anterior.	
4.Go-Live	Quan els resultats son els esperats, es passa a aquesta fase, en la que el projecte estaria llest per la possible posada a producció.	
5.Memoria	Durant tot el desenvolupament del projecte s'ha de anar redactant una memòria amb tots els passos, descobriments, canvis, funcionalitats i resultats del projecte, per a tal de que l'experiment es pugui reproduir en un futur.	



## 4.2. Detalls de les fases del desenvolupament

Per al projecte s'han establert unes fases per a tal de tindre una guia de com procedir en cada moment.

### 4.2.1. Preparació

#### Planificació

En aquesta fase es planifiquen les fases per a tal de adaptar-se a les dates de entrega esperades. Això permet també una millor visualització de la feina , temps requerit i temps restant.

#### Estudi de aplicabilitat

Primer s'ha de realitzar un estudi per a comprovar la viabilitat de la idea. Aquest valorarà tant la rentabilitat de la idea per a la empresa, com la del temps invertit per part del desenvolupador respecte al benefici que hi treguin.

#### Estudi de la tecnologia

Es necessari fer un estudi de la tecnologia disponible, tant a nivell hardware com software, es a dir, valorar fins a on es pot arribar amb el hardware disponible i amb el software que es farà servir.

#### Preparació de l'entorn

Per a desenvolupar el projecte es necessari un entorn de desenvolupament per a tal de poder comprovar si els resultats son satisfactoris. En aquest cas aquest consisteix en una maquina amb accés a la xarxa de la empresa.

### 4.2.2. Execució

#### Automatització de les aplicacions

Realització de les implementacions que permetran que les incidències es solucionin de forma automàtica.

#### Implementació del chatbot

Desenvolupament de la aplicació que s'encarregarà de respondre i gestionar les peticions dels clients.

#### Implementació del client web

Desenvolupament de la part web que serveix les peticions al *chatbot* i gestiona l'accés.

### 4.2.3. Avaluació

#### Comprovació dels resultats obtinguts

Avaluació dels resultats obtinguts durant la execució. Si aquests son incorrectes i no solucionen els problemes que havien de solucionar, s'ha de tornar al punt anterior per a seguir amb el desenvolupament.

### 4.2.4. Go-Live

#### Test amb els usuaris

Els usuaris realitzen una comprovació de la funcionalitat del projecte. Com en aquest cas els usuaris que fan el test no son els usuaris finals, es segueixen una sèrie de simulacions de incidències que poden diferir dels tests dels usuaris finals.

### 4.2.5. Memòria

#### Redacció de la memòria i resultats

Finalització del document de la memòria així com ampliació dels continguts amb imatges per a millorar la interpretació dels resultats del projecte.

#### Formació al equip

Per últim, es necessari formar a l'equip en tot el realitzat i après al llarg del projecte per tant que la informació es comparteixi i pugui ser útil en futurs projectes.

### 4.3. Valoració d'alternatives

Durant la realització del projecte poden sorgir desviacions del pla original que s'hauran d'analitzar gestionar per tal que l'impacte sobre la planificació inicial del projecte sigui el menor possible. Això s'aconseguirà amb la metodologia àgil utilitzada, ja que fent servir fites amb periodicitat curta aquestes desviacions es poden corregir de forma ràpida i senzilla.

- Fallada de servidor físic: La idea final es que a producció la maquina en la que estigui el chatbot sigui virtual, en aquest cas, la maquina es migraria a un altre servidor del clúster de producció.
- Fallada del servidor virtual: En cas de que la maquina virtual falli, estaria monitoritzat i hi hauria una alerta per a tal de poder actuar i trobar la solució lo abans possible. En cas de no trobar-se la solució es podria revertir a un estat anterior en el que funcionés.
- Caiguda elèctrica: El datacenter on estan situats els servidors virtuals te garantit un 99.982% de uptime anual.
- Caiguda de internet: Es disposa de una línia principal i de backup a totes les seus amb un uptime garantit de 99%.
- La incidència no permet ser automatitzada: Si una incidència no es pot automatitzar, aquesta es seguirà resolent com fins ara, mitjançant l'equip de IT
- Els usuaris no saben reconèixer el problema: Si els usuaris no saben com indicar quin es el problema que troben serà impossible que ho indiquin al chatbot, en aquest cas hauran de informar a l'equip de IT com en el cas anterior.

## 4.4. Diagrama de Gantt



## 4.5. Integració dels coneixements

Per a la realització de la estructura proposada al projecte s'han aplicat coneixements que he obtingut a base de observar les estructures de projectes existents a T2C.

Per a l'anàlisi de les incidències s'han aplicat els coneixements sobre tractament de documents apresos a la assignatura de CAIM

Per a crear el chatbot es necessari crear dues part de les que es compona.

La primera es la part de entrenament, per a això ha estat necessari aprendre com funciona la llibreria Keras, buscar models de dades i aprendre a llegir els models, aquí ha estat d'utilitat tots els coneixements de tractament de models apresos a la assignatura de APA.

Per a la part del predictor, es proposa una estructura que permeti enregistrar les converses mitjançant un registre en base de dades dels missatges i respostes, per a tal de poder mostrar a l'usuari una conversa consistent. En el predictor també s'incorpora un sistema de regles per a interpretar les peticions, sistema que s'implementa partint de un arbre de decisions i establint una sèrie de accions si es compleixen les condicions de una branca, per a la creació de aquest arbre han estat necessaris els coneixements establerts a TC. A part de l'arbre de decisions, en el cas de que cap dels casos encaixi amb la entrada donada, disposa de un predictor que troba una resposta possible a la entrada donada.

Tot el codi s'ha realitzat en Python, ja que, com s'ha comentat disposa de gran quantitat de llibreries per al tractament de llenguatge natural. Aquest llenguatge ja es familiar per els coneixements obtinguts en assignatures com ROB, CDI, CAIM, etc.

Finalment a la part del client web, s'ha hagut de aprendre com fer servir HTML per a tal de poder vincular-lo amb una api rest creada Python flask.

## 4.6. Identificació de lleis i regulacions

El desenvolupament de aquest projecte no està sotmès a cap llei o normativa. El projecte no modifica la estructura de la empresa ni les webs de les que disposa, ja que només incorpora un servei que actua de forma interna. Per tant, totes les ja existents segueixen sota les normatives pertinents, com serien la GDPR, cookies, condicions d'ús, etc. Segons es tracti.

## 5. Gestió econòmica i sostenibilitat

### 5.1. Gestió econòmica

#### 5.1.1. Consideracions i comentaris

Per desenvolupar aquest projecte, s'utilitzaran tots els elements esmentats en les entregues anteriors, ja que aquests tenen un cost. En aquest document s'incorpora l'estimació del cost del projecte tenint en compte els recursos humans, de *hardware*, *software*, llicències i costos indirectes.

El projecte no consta amb un pressupost establert, ja que el preu es negociaria després de la creació del prototip, per tant, s'ha de mantenir un registre acurat de les hores invertides per tal de poder ajustar el preu del producte el màxim possible.

Els elements tinguts en compte per a determinar els costos són agafats del diagrama de Gantt de l'entrega anterior.

#### 5.1.2. Pressupost de recursos humans

Aquest projecte és portat només per una persona que s'ocuparà de tots els rols del projecte: cap de projecte, dissenyador, programador i beta tester. A la següent taula anotem el nombre d'hores que es requerirà de cada rol per tal de poder assignar un valor econòmic a aquestes.

Rol	Hores	Preu per hora	Preu total
Cap de projecte	140 h	50€/h	7000€
Dissenyador	40 h	35€/h	1400€
Programador	240 h	35€/h	8400€
Beta tester	80 h	30€/h	2400€
TOTAL	500 h		19,200€

### 5.1.3. Pressupost de hardware

Per la implementació del projecte, en la part *hardware* no serà necessària l'adquisició d'elements addicionals, donat que l'empresa pot proveir de tots els dispositius necessaris.

### 5.1.4. Pressupost de software i llicències

Pel projecte es procurarà fer servir *software* lliure i gratuït o que ja es tinguin les llicències adquirides per això es pot dir que el cost de *software* i de llicències també seria nul.

### 5.1.5. Despeses indirectes

En tot projecte d'informàtica existeixen unes despeses derivades de l'ús d'electricitat o de paper, entre altres.

Producte	Preu	Unitats	Cost Aproximat
Electricitat	0,12€/kWh	60000 kWh	7.200€
Paper	7,90€/pack	1 pack	7,90€
TOTAL			7.207,90€



### 5.1.6. Pressupost total

Un cop analitzats tots els costos podem establir el cost total de l'elaboració del projecte.

Concepte	Cost aproximat
Recursos humans	19,200€
Hardware	0€
Software i Llicències	0€
Costos indirectes	7.207,90€
TOTAL	26,407.90€

Podem observar que el preu és competitiu, tot i que es podria valorar la utilització de serveis de *chatbots* externs per tal de reduir el cost d'elaboració a canvi de pagar una llicència pel recurs adquirit.

A part del cost d'elaboració s'ha de concretar amb el client un preu de manteniment del producte però aquest es decidirà un cop aquest adquireixi el producte.

### 5.1.7. Control de desviacions

Un dels principals problemes que pot sorgir i afectar el pressupost és la desviació en alguna de les tasques del projecte. Per evitar aquestes desviacions s'utilitzarà el diagrama de Gantt per seguir el flux de temps establert i poder acabar el projecte dins dels termes pactats.

## 5.2. Àrea social

Aquest projecte es troba dins del món del IT, més específicament dins del manteniment de sistemes. El món del manteniment de sistemes, és un món on es requereix molts recursos humans, molt sovint destinats a tasques repetitives i monòtones.

L'objectiu d'aquest producte és que aquest recurs quedi alliberat d'aquestes tasques, per tant, per un costat és capaç de millorar la productivitat d'aquests treballadors. En canvi, per part del client final, és a dir, del comprador que utilitzarà aquest servei en lloc dels empleats de manteniment de sistemes aconseguirà que moltes d'aquestes incidències es resolguin sense cap temps de demora. Per tant, el client també podrà ser capaç d'incrementar la seva productivitat.

## 5.3. Àrea ambiental

Pel que fa a l'àrea ambiental, aquest projecte no té una gran influència més que els consums indirectes i els seus mètodes d'obtenció. També es podria considerar una despesa de recursos si es fes servir un entorn físic pel projecte però com s'ha optat per un entorn virtual, aquesta petjada al medi ambient també s'ha evitat.

## 5.4. Puntuació sostenibilitat

Pel que fa a la puntuació de sostenibilitat del projecte podem classificar-lo amb les següents puntuacions en cadascun dels diferents àmbits:

Sostenible?	Econòmica	Social	Ambiental
Puntuació	8	9	8

## 5.5. Conclusió sobre la sostenibilitat

Podem extreure que perquè un projecte sigui sostenible ha de ser capaç d'aportar millores a la major quantitat de persones possibles fent servir sempre els mínims recursos possibles.

Un cop establert que es considera sostenible podem remarcar els principals punts a favor i en contra del projecte:

A favor:

- Millora de productivitat en client: El client podria aconseguir un estalvi de temps que aplicaria a altres tasques.
- Millora de productivitat al tècnic de sistemes: El tècnic podria aconseguir un estalvi de temps que aplicaria a altres tasques.
- El cost de manteniment de l'eina es pràcticament nul: Només requereix de revisió eventual en cas de detectar-se *bugs* futurs.
- Nul·la empremta ecològica: A l'estar en un entorn virtual, l'empremta ecològica de l'eina es torna nul·la.

En contra:

- Dificultat en la integració: Al començament de la integració de l'eina en l'entorn de treball del client és possible que aquesta comporti una complicació al moment de solucionar les incidències per la falta de costum, el que pot portar a una pèrdua de productivitat temporal.
- Al tractar-se d'un projecte específic per uns serveis determinats es redueix l'eficiència que podria tenir l'eina, per tant, es pot dir que s'estaria desaprofitant tot el potencial de l'eina.

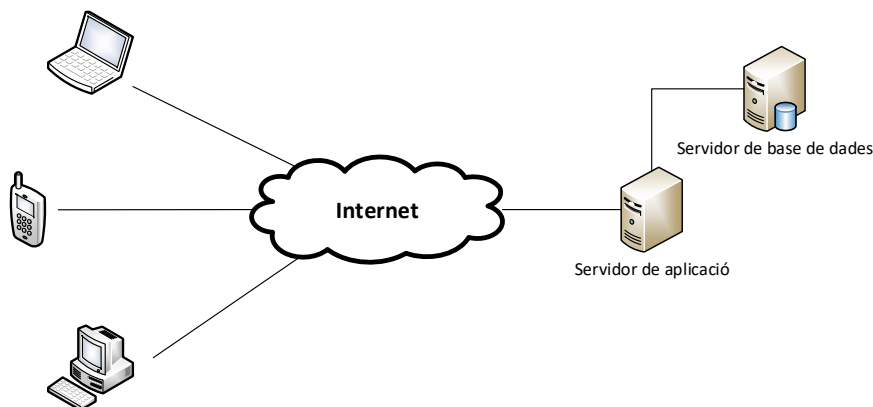
Per tant, podem concloure que aquest projecte, es pot considerar com a sostenible, ja que per les millores en productivitat que pot aportar al client i al treballador de manteniment de sistemes, compensa de bon tros el possible cost que tindria el manteniment d'aquesta eina. També es pot dir que el projecte només afectaria a la sostenibilitat en l'àmbit social i econòmic i que en l'àmbit ecològic no hi entraria.

## 6. Disseny

### 6.1. Detalls de la infraestructura dissenyada

#### 6.1.1. Descripció de la infraestructura

Per la realització del projecte s'ha optat per un disseny d'aplicació que faci servir un sistema client-servidor, en el qual el servidor s'encarregarà de fer servir les peticions dels usuaris dels clients. Aquest disseny concorda amb el sistema utilitzat a l'empresa per les diferents aplicacions, en què es dedica una màquina virtual en la qual es construeix l'entorn de treball i, per mitjà d'un client s'accedeix al servidor.



*Il·lustració 1 esquema de la infraestructura*

Com es pot observar a la figura, la infraestructura disposa de diversos *nodes*. Aquest però haurien d'anar en màquines separades i virtuals, per tal de facilitar-ne el manteniment, tot i que per aquest projecte s'han incorporat tots els *nodes* en una mateixa màquina física.

#### Node servidor

Aquesta màquina és la que s'encarrega de realitzar el processat de les peticions dels clients i d'enviar la resposta al client. Es tracta d'una màquina orientada al processament, en la qual s'hauria de prioritzar un sistema operatiu *Unix* per les opcions que dóna a l'hora de configurar serveis.

### Node de base de dades

Aquesta màquina és la que s'encarrega d'emmagatzemar dades per tal que el servidor pugui proporcionar un històric als clients (com seria l'historial de xat). Per aquest node s'ha de proporcionar un equip amb un sistema de base de dades en què es pugui emmagatzemar la quantitat d'històric acordada amb el client. Per això, es tracta d'un sistema dedicat a l'emmagatzematge de dades.

Per aquesta màquina i l'anterior hi ha una altra condició important per termes de funcionament de l'aplicació que és el temps d'uptime de la màquina, quant més alt sigui aquest, més estona estarà la nostra aplicació en funcionament i disponible.

### Node client

La idea del servidor client és que sigui accessible des del màxim nombre de plataformes, per això s'ha optat per un client web, ja que en aquests moments qualsevol tipus de dispositiu es capaç de accedir a un navegador web i introduir-hi una url.

## 6.1.2. Descripció de les tecnologies potencials

Per al desenvolupament del projecte, s'ha tingut en compte els diferents entorns existents a la empresa i les diverses necessitats que es troben els clients per a això s'ha creat una aplicació que sigui capaç de ser posada a producció en la major quantitat de escenaris possibles, això vol dir, amb els mínims recursos necessaris i sobre la majoria de sistemes operatius.

Tot i haver pensat en aquestes característiques, per a l'entorn del projecte s'ha fet servir un equip de sobretaula de gama mitja-alta amb un sistema operatiu Windows 10, aquest sistema operatiu i equip s'han escollit en primer lloc per disponibilitat i per dificultat en la preparació de entorns en aquest sistema.

Per últim, el motiu per el que s'ha muntat en un equip de sobretaula ha estat per a poder veure que no es necessari un equip dedicat i de alt rendiment per a poder executar la aplicació.

## 6.2. Disseny del chatbot

Per a poder realitzar una eina que sigui escalable, fiable, i útil, es pretén desenvolupar un chatbot amb la estructura dissenyada.

Per a que aquest sigui accessible des de el major numero de plataformes possibles es decideix que la forma de comunicació sigui mitjançant un client web o pagina web:

En quan al chatbot en si, tindrà les següents característiques:

- Es tractarà de un chatbot amb un disseny híbrid de dues capes, una amb un arbre de decisions i per sota una capa que es servirà de una xarxa neuronal.
- L'arbre de decisions permetrà de forma senzilla gestionar els diferents casos demanats, així com afegir-ne de nous en cas de ser necessari, tot això sense ser necessari un reentrenament del model, i per tant sense ser necessaris coneixements profunds sobre chatbots. Aconseguint així que qualsevol membre de l'equip pugui afegir la acció amb el mínim coneixement de la acció a implementar.
- La capa que s'executarà per sota, es a dir, la xarxa neuronal, donarà resposta als usuaris per a les peticions que no s'ajustin a les indicades en la capa superior. Aquesta capa, serà necessari entrenar-la mínim un cop, un cop entrenada es pot exportar i deixar llesta per a fer servir.
- Per a poder tindre un registre de les accions realitzades per el chatbot, es manté un registre de converses en base de dades, aquest registre pot servir per a poder, més endavant, fer un estudi de quins son els casos de us més empleats o per altre banda, si hi algun cas que no s'utilitzi.
- Per tal de que la eina no es faci servir de forma irresponsable, s'estableix un sistema de login per a tal de identificar l'autor de cada petició.

## 7. Implementació

### 7.1. Entorn

L'entorn que s'ha fet servir per al desenvolupament es una maquina de sobretaula de gamma mitja-alta amb un sistema operatiu Windows 10. La màquina feta servir no te perquè ser igual a les de l'entorn de producció, ja que la aplicació es capaç de executar-se en diverses plataformes. Tot i això ens els següents passos es detallarà com realitzar la instal·lació en l'equip utilitzat.

### 7.2. Instal·lacions i configuracions inicials

#### 7.2.1. Instal·lació del sistema operatiu

Primer, per a la preparació de l'entorn, hem de instal·lar el sistema operatiu, com la aplicació es capaç de funcionar en gran part de sistemes operatius, independentment de les característiques de aquest, es per això que la fase de instal·lació no es detallarà, ja que les característiques no han de ser específiques.

#### 7.2.2. Instal·lació de python

Per poder fer que l'aplicació executi, es necessari instal·lar la versió de *Python 3.6*, aquest és el llenguatge en el qual s'ha implementat l'aplicació, fent servir d'altres llibreries que s'explicaran més endavant.

Per la instal·lació de *Python* en *Windows*, es necessari accedir a la pàgina oficial, i descarregar-se l'instal·lador. Com a complement de *Python*, també es recomanable instal·lar l'eina *pip*, que és el gestor de *packages* de *Python* i facilita la instal·lació de les llibreries.

### 7.2.3. Instal·lació de llibreries

Per l'aplicació són necessàries certes llibreries de *Python*, cadascuna té la seva funció, i el seu lloc on són necessàries, per això cal instal·lar-les abans d'executar el codi. Les llibreries a instal·lar són les següents:

Flask: llibreria que permet implementar programes *Python* amb l'estructura d'una *Api rest*, estructura necessària pel disseny de client servidor que s'ha implementat.

Spacy: llibreria de tractament de dades en llenguatge natural, feta servir pel reconeixement de normes en l'arbre de decisions.

Re: llibreria de tractament d'expressions regulars per la identificació de característiques.

Keras: llibreria per tractament de dades amb *deep learning*, que facilita el treballar amb xarxes neuronals. Necessària en la implementació del model de chatbot.

Nltk: Aquesta llibreria, permet treballar amb llenguatge natural, i així processar-lo d'una forma més usable. Necessària en el tractament de dades per l'entrenament del chatbot.

Sklearn: Es tracta d'una llibreria amb múltiples eines per treballar amb anàlisis de dades i mineria de dades. En el projecte, s'utilitza per realitzar la distribució de dades al separar els models de entrenament i de test.

També s'han instal·lat llibreries menys necessàries per facilitar l'obtenció d'informació. Les quals són necessàries a nivell d'automatització d'incidències i no influeixen en el funcionament del chatbot.

## 7.3. Automatització de incidències

Per a poder fer que el chatbot sigui capaç de realitzar accions, es necessari implementar el com les accions s'han de realitzar, per a això, el primer pas ha estat veure quines accions havia de ser capaç de realitzar el chatbot.

Alhora d'escollir quines accions havia de ser capaç de realitzar el chatbot d'inici la empresa va prioritzar la obtenció de informació de diverses eines que ja tenim, i la resolució de incidències freqüents, que fossin simples, i per a una primera fase, que no afectessin massa als entorns i processos principals del client.



Com a resultat de buscar accions a automatitzar amb aquestes característiques, les accions a realitzar es van reduir bastant, i de aquestes se'n van escollir les següents per a implementar la automatització.

### 7.3.1. Obtenció de informació

La primera acció que es va decidir automatitzar va ser obtenció de informació sobre l'estat dels servidors. Per a aquest procés, la empresa disposa de una eina ja en funcionament, Zabbix, aquesta eina s'encarrega de realitzar el monitoratge dels servidors, tant virtuals com físics. Per tant, per a automatitzar la acció de obtenir aquesta informació, s'ha decidit aconseguir extreure la informació de la que disposa aquesta eina i servir-la a l'usuari. Per a això el que s'ha fet servir, es la Api que proporciona aquesta eina, que es capaç de extreure la informació de les maquines sempre i quant el nom de aquesta sigui conegut. Per a això, s'han definit uns requisits en l'arbre de decisions per a poder entrar en aquesta branca, aquests son: identificar un servidor, determinar una intenció de obtindre informació de aquest.

Una altre acció a automatitzar, ha estat la obtenció de l'estat en el que l'usuari te les incidències assignades. Per a això, el client, fa servir una eina anomenada Clio, que es tracta de una eina de ticketing, en la que es creen "Clios" (tiquets) i s'assignen als diferents usuaris, per a tal que aquests en siguin conscients, i puguin solucionar-los. Aquesta eina, com la anterior, disposa de una api que permet obtenció de informació, que es el que s'ha utilitzat per a fer la obtenció de informació. En aquest cas, per a poder obtindre la informació, es necessari donar el nom de usuari del que es vol utilitzar la informació. Aquesta informació, com en el cas anterior, s'han establert com a condicions per a poder entrar a la branca de l'arbre de decisions.

Una informació que es volia ser capaç de obtenir, era l'estat dels processos que es s'executen de forma automàtica. Aquests s'executen fent servir una eina que s'anomena VisualCron, que es un scheduler que permet organitzar execucions, i definir-ne horaris i condicions. Aquesta eina també disposa de una api, aquesta permet obtindre els estats de les tasques, així com aquelles que les execucions hagin fallat. Alhora de obtindre la informació s'ha decidit extreure nomes les tasques que han fallat, ja que el volum de tasques en execució pot arribar a ser molt elevat i podria ser carregant per a l'usuari, i la Api no permet filtrar per tasques a buscar de forma

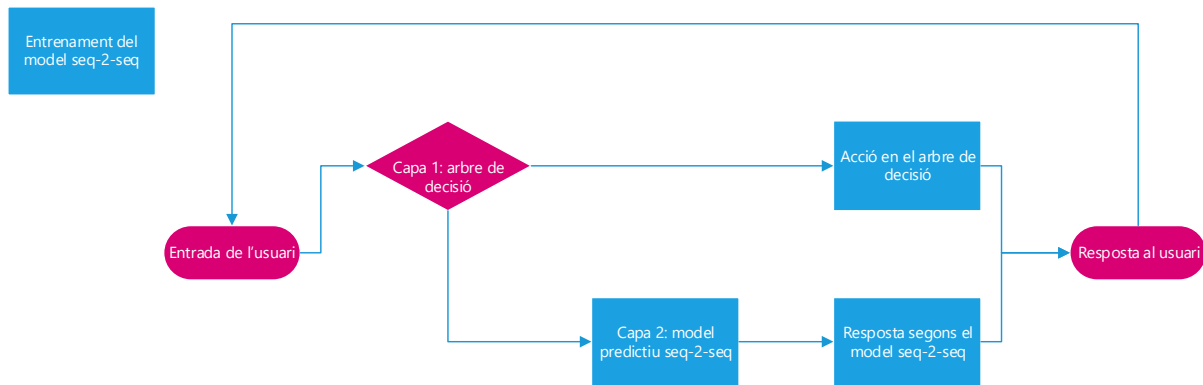
fàcil, es necessari saber el nom exacte de la tasca, i aquesta es una informació que l'usuari final podria no saber. Per tant s'ha optat per que es pugui consultar només les tasques que hagin fallat. En aquest cas per a la obtenció de informació no es necessita ninguna informació addicional, com seria l'usuari o nom de la tasca. Per això, s'ha establert que l'usuari indiqui la intenció de obtindre l'estat de les tasques de VisualCron, (o acrònim o diminutiu dels mes utilitzats). Aquest requeriment s'ha establert per a tal de que així el chatbot no acabi confonent ordres.

### 7.3.2. Resolució de incidències

Finalment, com a resolució de incidències, tenint en compte els requeriments de la empresa de no ser invasius a cap entorn i evitar afectar a entorns de producció la llista de incidències possibles a solucionar, s'ha acurtat molt. Es per això que les dues incidències a automatitzar, van ser primer, el desbloqueig de usuaris, aquesta incidència, es pot solucionar amb un script, que realitza un accés al servidor on es troba el active directori de la empresa i llença les comandes en bash per a desbloquejar l'usuari. Per a aquest script, es necessari detectar el nom d'usuari, i en el cas de passar l'script a productiu, afegir un control de seguretat, com demanar la contrasenya.

## 7.4. Implementació del chatbot

Per al chatbot, s'ha implementat un disseny híbrid, que combina dues capes, la primera, un sistema de normes que permeten identificar una sèrie de estats definits, que cadascun implica un comportament específic, definits en l'apartat anterior. Per a aquest, s'ha dissenyat un arbre de decisions que amb un anàlisi de la entrada guia a la branca corresponent. Per als casos que no s'encaixi en cap d'aquestes condicions, la entrada es processaria per la segona capa, que es tracta de un chatbot realitzat amb una adaptació del model seq-2-seq per a traducció de text.



Il·lustració 2 Diagrama de flux del chatbot

A la figura es pot veure un esquema de com procedeix la execució del chatbot, que consta de dues parts, l'entrenament i la predicció, i dins de la segona, trobem subfases definides per les diverses capes, que s'executen de forma seqüencial.

#### 7.4.1. Primera capa: Arbre de decisions

Per a aquesta primera capa del chatbot, no es necessari implementar cap fase de entrenament, si més no, es necessari realitzar el preprocessament de la entrada. Per a aquesta part, es necessària la llibreria spacy, que permet de forma senzilla realitzar un tractat de un text en llenguatge natural. També es necessària la creació de l'arbre de decisions.

##### Preprocessament de la entrada

Per al preprocessat de la entrada, s'han tingut en compte les característiques que es buscaven per a les accions implementades en la fase anterior, de aquesta han sorgit diversos tipus de condicions per a cada acció, com serien, la detecció de un username, el nom de un servidor i la detecció dels noms de les eines a les que fa referencia la acció. Aquest reconeixement s'ha realitzat amb la creació de expressions regulars, una per a cada característica, en el cas del servidors i els usuaris, s'han tingut en compte la nomenclatura que s'ha fet servir fins al moment en l'assignació de aquestes. En canvi per al nom de les eines, s'ha establert el reconeixement de les diferents formes de referir-se a aquestes, siguin acrònims o els noms més usats per a aquestes.

Un cop creades les expressions regulars per a la detecció de les característiques, es necessari examinar la entrada per a analitzar les intencions.

Per a aquest punt son necessàries diverses coses, el primer, realitzar un tokenitzat de la entrada, per a poder escanejar cada token i veure si encaixen en les característiques que cerquem. Un cop fet l'escanejat, es treuen les característiques generals de la entrada, per a poder avaluar si coincideixen amb les esperades, també per a un possible us en el futur, s'extrauen les entitats detectades en la entrada, però en els casos que analitzem no son necessàries.

#### Presa de decisions

Alhora de realitzar la decisió, s'ha implementat un arbre lògic, en el que segons les característiques existents, s'entra per una o altre branca, i en cas de no trobar-ne cap que encaixi en el detectat, es procedeix a deixar la resposta a la segona capa. Per a tal de augmentar la efectivitat de aquesta primera capa de filtre, s'ha implementat un sistema de comprovació de similitud de frases. En el que mitjançant una funció de la llibreria spacy, anomenada similarity, avalua quan semblants son dues frases amb un valor de 0 a 1. En aquesta comprovació, s'ha establert una similitud que va del 0.8 al 0.9 depenent de la entrada i el nombre de característiques necessàries per a la acció en qüestió.

En cas de que aquesta primera capa no sigui capaç de definir una acció per a la entrada, la resposta, procedirà a deixar-se a mans de la segona capa.

#### 7.4.2. Segona capa: seq2seq

Aquesta segona capa, requereix de mes preparació que la primera, ja que es tracta de una capa que treballa amb un model predictiu. Per això, s'ha de definir una fase de entrenament prèvia a la utilització del chatbot.

#### Entrenament

Abans de poder realitzar l'entrenament del model, s'ha hagut de decidir el tipus de model que es volia aconseguir, el tipus de dades que es faria servir per a entrenar-lo i les característiques que serien necessàries. El tipus de model per el que s'ha optat finalment, es un model de tipus seq-2-seq, que fa servir una xarxa neuronal de tipus LSTM (Long Short Term Memory) per a poder aprendre del client amb el temps.

En quant al model de dades que s'ha fet servir ha estat GloVe [\[17\]](#), un sistema de vectors preentrenats que redueixen el temps de entrenament i de tractament de dades.

Per a la implementació del chatbot s'ha utilitzat la llibreria Keras, una llibreria per a treballar amb xarxes neuronals a alt nivell, amb integració amb plataformes com ara TensorFlow.

Per a l'entrenament del model, s'han implementat el següents passos:

1. Lectura dels fitxers de entrada o descarrega de aquests: El primer pas, es obtenir el conjunt de dades que es farà servir per a entrenar el model, en cas de no tindre cap al path esperat, el programa procedirà a descarregar un dels accessibles a la pagina dels desenvolupadors de GloVe (universitat de Stanford)
2. Tractar les dades per a convertir-les a dades valides per a introduir al model: Les dades, a l'inici venen en un format determinat, que s'ha de modificar per a poder introduir-les al model amb Keras, això implica tokenitzar-les, indexar-les, definir relacions pregunta-resposta i extreure'n el context.
3. Crear el model: Per a tal de poder crear el model, un cop tenim les dades en un format apte, es fa servir la llibreria Keras per a poder obtindre la xarxa neuronal que servirà per a definir el model. Per a aquest pas, es fan servir les crides Input i LSTM que permeten crear Tensors utilitzats en la creació del model.
4. Entrenar el model: Es defineixen un conjunt de test i un de entrenament i es generen de forma aleatòria, aquests conjunts es defineixen amb la llibreria sklearn, que retorna els índexs a utilitzar per a la distribució. Un cop separats els conjunts s'entrena el model fins al punt definit amb els sets de dades donats.
5. Guardat de les dades: El model, s'emmagatzema a disc cada x nombre de passos definits al programa, això permet executar el programa a mig entrenament, alhora que ha facilitat el poder implementar la fase del predictor sense haver de esperar a un primer entrenament del model per complet.

## Predicció

Per a realitzar el predictor, al voler fer-lo ser un programa que fos accessible mitjançant una Api, s'ha tingut en compte que no tingues cap us innecessari de variables ni un ús de memòria que augmentes al llarg de la execució. Això es perquè en productiu, el programa pot estar en execució durant dies seguits sense que la aplicació es pari, i un augment crític dels recursos utilitzats podria causar que la resta de aplicacions situades en el mateix equip, si es que n'hi ha, funcionessin malament o fallessin.

Tenint en compte aquestes mesures, el predictor s'ha implementat de tal forma que sigui una classe aïllada a la api, es a dir, la api incorpora el predictor com a una component invariable a la que nomes pot accedir mitjançant crides.

Un cop definida la forma en la que la api interactua amb el chatbot, s'ha implementat el predictor de la següent manera:

1. Inicialització i carrega del model: Per a tal de poder obtenir el model entrenat el la fase anterior, es necessari carregar els fitxers guardats, per això es carrega el model en la instancia creada del predictor. Per a això, la llibreria Keras esta preparada per a realitzar la importació del model.
2. Un cop inicialitzat el model, es programa la part que s'encarrega de proporcionar una resposta a la entrada:

Aquesta crida, te dues fases, la primera en la que es processa la entrada, en aquest cas, això consisteix en, primer, tokenitzar la entrada, un cop tokenitzada, trobar a quin estat del model pertany la entrada, i un cop es troba l'estat de la entrada, es procedeix a predir quin es el següent estat per a la entrada donada.

La segona predicció es realitza de forma continuada fins que es troba un indicador de fi de predicció o s'arriba al límit establert manualment.

## 7.5. Implementació de la interfase web

El desenvolupament de la interfase web combina 3 llenguatges de programació de pàgines web: *HTML*, *CSS* i *Javascript*.

L'estructura bàsica, és a dir, l'esquelet de la web, es realitza mitjançant les funcions típiques d'*HTML*. A través de *Javascript* es creen les estructures de grandària dinàmica, com són els contenidors dels textos de resposta. Per a la interacció del

client amb el servidor s'utilitza una api python, que permet integrar crides a una Api amb Flask.

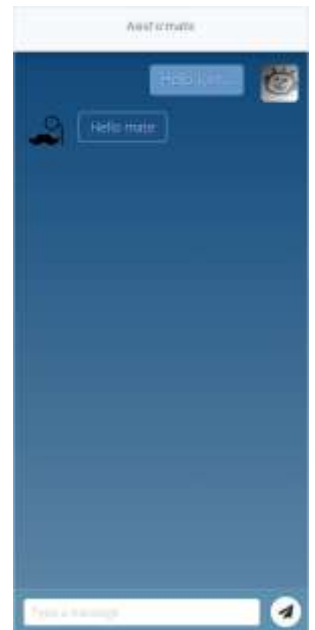
Mitjançant una fusió entre la integració de javascript i les crides amb Flask, es crea la estructura dinàmica, com seria la conversa, incloent-hi missatges de sessions anteriors.

Al tractarse de una conversa en viu, la pagina es genera de forma dinàmica, obtenint es incorporant a la pantalla mostrada les respostes de la api.

S'ha implementat una pantalla de login de forma temporal, ja que a la posada a producció es faria servir el sistema de usuaris del client, en cas de disposar-ne de un.



*Il·lustració 3 Disseny gràfic de la interfase web*



Aquí es mostra la pantalla en la que es mostra la conversa amb el chatbot, a la dreta es poden veure els missatges enviats per l'usuari (seguint el patró que fan servir altres aplicacions de missatgeria, com WhatsApp, Telegram o Facebook Messenger). El disseny de la estructura s'ha construït pensant en una adaptació a tot tipus de dispositius, tant mòbils, tabletas com navegadors, per això es tracta de un disseny minimalista i auster.

## 7.6. Comunicació web-chatbot

Per a tal de que la web i el chatbot es comuniquin ha estat necessari la adaptació del sistema de entrada-sortida del programa per a que aquest s'integri amb la web. El primer pas per a això ha estat convertir el chatbot a una estructura de api rest, es a dir, les crides a la predicció de respostes, s'ha implementat de tal forma que mitjançant una crida post es pugui obtenir la resposta donada per el chatbot. Per a poder fer això, s'ha requerit de la llibreria Flask de python, que permet crear funcions que s'invoquen com a crides de una Api Rest, amb aquest mètode s'aconsegueix el poder fer crides al chatbot de forma externa, sempre i quant la aplicació estigui executant-se. Addicionalment, s'afegeix també a la web crides que permeten recuperar les dades de les que disposa la aplicació del chatbot.



## 8. Conclusions

Quan es va proposar en la empresa realitzar aquest prototipus, van sorgir dubtes si en lloc de un chatbot, es podien fer servir eines mes senzilles per a la implementació del sistema, com podia ser un sistema de menús que permetessin realitzar la mateixa funció, no es tenia clar quin era l'enfoc que se li havia de donar al projecte.

Els chatbots, eren un camp que dins de la empresa no s'havia fet servir encara, només com a eina, sense haver arribat a desenvolupar-ne cap. Es per això que és va acabar optant per aquest tipus de sistema, per a poder aprendre programació que apliqués models de intel·ligència artificial, i així més endavant, poder avançar en aquest àmbit.

Un cop acabat aquest projecte, puc afirmar que seguir amb el disseny escollit va ser la millor decisió, ja amb la implementació de aquest disseny s'han trobat possibles millores per a un futur, apart de una nova oportunitat en quant a la oferta de productes de la empresa.

Tot i que el projecte no aporta res nou al món del desenvolupament de chatbots, permet que la empresa disposi de un producte propi, sense cost addicional i sobretot, totalment editable.

Personalment, aquest projecte m'ha aportat la experiència de treballar amb un model complex al llarg de tot el procés, des de la cerca de dades, fins a la extracció de les dades del predictor, fins a ferles arribar al client de forma útil.

Cal fer incís en la gran quantitat de investigació que té el projecte al darrere, ja sigui per la part de buscar la forma de automatitzar les incidències, la cerca de models i opcions per a poder implementar el chatbot, i sobretot, la cerca de necessitats del client, per a poder fer del chatbot, una eina útil des de la primera versió.

## 9. Treball futur

El chatbot, encara esta en una fase en la que pot ser una eina molt mes completa, tant sigui augmentant la integració amb la resta de entorns, com millorant les capacitats del chatbot, així com implementant un sistema de multi-llenguatge. Es per això que es proposa la millora de les tècniques del chatbot i la expansió a un sistema que pugui implementar un sistema de múltiples llenguatges.

Durant l'anàlisi de les incidències, es van valorar moltes incidències que per a aquest projecte es van ometre, ja fos per falta de recursos, o per poca freqüència de aquestes. Es per això, que en acabar el projecte, l'arbre de decisions hauria de seguir en expansió, ja sigui, integrant més plataformes de les que extreure informació, com integrant noves incidències a solucionar. Cadascuna de aquestes amb les seves característiques que permetin identificar-les. Aquesta es una feina que s'haurà de mantenir de forma continua durant tot el temps de vida útil de la eina per a evitar que aquesta caigui en desús.

A nivell de la interfase web, existeixen millores que es podrien implementar, com seria la integració del sistema de login amb el de la empresa.

La quantitat de funcionalitats que es poden afegir a la eina pot semblar infinita, i certament així podria ser, ja que les possibilitats que té son tan amples que nomes queda limitat per el que l'entorn en el que es desplegui i les especificacions del client que tinguin possibilitat d'automatització i integració.

## Referències

1. Géron, A., 2017. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems.* "O'Reilly Media, Inc.". [en linea] Disponible a:  
<https://books.google.es/books?hl=en&lr=&id=khpYDgAAQBAJ&oi=fnd&pg=PP1&dq=Hands-on+Machine+Learning&ots=kLCBNMCTm3&sig=klpEuy7IfqdZLSFIKOIqV7kFrQ>
2. McCulloch, W.S. and Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), pp.115-133. [en linea] Disponible a: <https://link.springer.com/article/10.1007/BF02478259>
3. Rosenblatt, F., 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6), p.386. [en linea] Disponible a: <http://psycnet.apa.org/record/1959-09865-001>
4. Sepp Hochreiter and Jürgen Schmidhuber. *Long short-term memory.* *Neural Comput.*, 9(8): November (1997). [en linea] Disponible a: <https://www.bioinf.jku.at/publications/older/2604.pdf>
5. Chan, Y.H. and Lui, A.K.F., 2018, May. Encoding emotional information for sequence-to-sequence response generation. In *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)* (pp. 113-116). IEEE. [en linea] Disponible a: <https://ieeexplore.ieee.org/abstract/document/8396177/>
6. Van Landeghem, J., 2016. Sequence-to-Sequence Learning for End-to-End Dialogue Systems. [en linea] Disponible a: [https://www.researchgate.net/profile/Jordy\\_Van\\_Landeghem/publication/313375963\\_Presentation\\_on\\_end-to-end\\_results\\_of\\_the\\_project/data/589884beaca2721f0daf3173/February06.pdf](https://www.researchgate.net/profile/Jordy_Van_Landeghem/publication/313375963_Presentation_on_end-to-end_results_of_the_project/data/589884beaca2721f0daf3173/February06.pdf)

7. Eleconomista.es. (2016). *¿Qué son exactamente los chatbots y para qué sirven?* - *elEconomista.es*. [en línia] Disponible a: <http://www.eleconomista.es/tecnologia/noticias/7488529/04/16/Que-son-exactamente-los-chatbots-y-para-que-sirven.html>.
8. 40deFiebre. (2018). *¿Qué es un chatbot?*- *Socialmood* [en línia] Disponible a: <https://www.40defiebre.com/que-es/chatbot/>.
9. Python.org. (2018). *Welcome to Python.org*. [en línia] Disponible a : <https://www.python.org/>.
10. Mysql.com. (2018). *MySQL*. [en línia] Disponible a: <https://www.mysql.com/>.
11. Html.com. (2018). *Home*. [en línia] Disponible a: <https://html.com/>.
12. W3schools.com. (2018). *CSS Tutorial*. [en línia] Disponible a: <https://www.w3schools.com/Css/>.
13. TensorFlow. (2018). *TensorFlow*. [en línia] Disponible a: <https://www.tensorflow.org/>.
14. Keras.io. (2018). *Keras Documentation*. [en línia] Disponible a: <https://keras.io/>.
15. Anon, (2018). *spaCy · Industrial-strength Natural Language Processing in Python*. [en línia] Disponible a: <https://spacy.io/>.
16. Insomnia.rest. (2018). *Floating Keyboard Software*. [en línia] Disponible a: <https://insomnia.rest/>.
17. Pennington, J., Socher, R. and Manning, C., 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543). [En línea] Disponible a: <http://www.aclweb.org/anthology/D14-1162>

18. Docs Google. Cuestionario de Estudiantes de Ingeniería Informática. [En línia]  
[https://docs.google.com/forms/d/e/1FAIpQLSelZixKIUFbCn1oVkd2JM3yxCc208E85\\_RgKZclKd8eUu3GvBg/viewform](https://docs.google.com/forms/d/e/1FAIpQLSelZixKIUFbCn1oVkd2JM3yxCc208E85_RgKZclKd8eUu3GvBg/viewform)
19. EL INFORME DE SOSTENIBILIDAD DEL TFG. [aut. libro] Professorat de GEP.  
UPC. Pàgina 3. Figura 2.
20. Abdellah, Z., Ahmadi, A., Ahmed, S., Aimable, M., Ainscough, R. and Almeida, J., 2004. International human genome sequencing consortium. *Nature*, 409, pp.860-921. [en línia] Disponible a: <https://www.nature.com/articles/nature03001>
21. ENCODE Project Consortium, 2012. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414), p.57. [en línia] Disponible a: <https://www.nature.com/articles/nature11247>
22. Forbes, 2017, *Illumina Promises To Sequence Human Genome For \$100 -- But Not Quite Yet* [en línia] Disponible a:  
<https://www.forbes.com/sites/matthewherper/2017/01/09/illumina-promises-to-sequence-human-genome-for-100-but-not-quite-yet/#4a567e8c386d>
23. Pal, S.K. and Mitra, S., 1992. Multilayer Perceptron, Fuzzy Sets, Classification. [en línia] Disponible a: <http://library.isical.ac.in:8080/jspui/bitstream/10263/4569/1/308.pdf>
24. Abraham, A., 2005. Rule-Based expert systems. *Handbook of measuring system design*. [en línia] Disponible a:  
<https://onlinelibrary.wiley.com/doi/book/10.1002/0471497398>
25. Huang, J., Zhou, M. and Yang, D., 2007, January. Extracting Chatbot Knowledge from Online Discussion Forums. In *IJCAI*(Vol. 7, pp. 423-428). [en línia] Disponible a: <http://www.aaai.org/Papers/IJCAI/2007/IJCAI07-066.pdf>
26. Shawar, B.A. and Atwell, E., 2002. *A comparison between Alice and Elizabeth chatbot systems*. University of Leeds, School of Computing research report 2002.19. [en línia] Disponible a: <http://eprints.whiterose.ac.uk/81930/>